Approach for Modelling and verification of flexible and adaptable businesses processes

MADANI MOHAMED [1], BEKKI KHADHIR[1]

[1]départemenr de l'informatqiue, université ibn khaldoun tiaret
Mohamed_madani@yahoo.fr, Kh_bekki@yahoo.fr

Abstract*— Nowadays, the business processes need to be more flexible, adaptable and secured. These three criteria increase the performances of applications inter organizations and guarantee their stability. For this purpose, we plan to re-examine the way of modeling the business processes so they will flexible and adaptable. For that, as for the adaptation, we adopt separation between concerns. We separate the development of the functionality concern from the transversal concerns (eg: security, context). As for flexibility, we propose a new model of description of the business processes based on ECA rules (Event -Condition-Action). So, our approach of modeling the business processes is an multi concerns approach (Security, I nteraction) based on our formalism suggested CECAETE (Concern, Event,Condition, Action, check Execution, Time, else Event). First, we govern any business rule as a CECAETE rule. Then, to verify the rules based business process, we build a graph of rules. This graph is based on the relationships between the rules of the same concerns and of different concerns. In this paper, we discuss also, the formal verification of the CECAETE rules based business process.

KEY WORDS : FLEXIBLE MO DELING , BUSINESS PROCESS, THE SEPARATION OF ASPECTS, BUSINESS RULES, SECURITY , ECA RULES

1. INTRODUCTION

The service-oriented architecture (SOA) allows collaborating and sharing critical data. Web services are explicit software units that can through their interfaces to be described, published, and most importantly, composed (dynamically) using XML-based protocols (for example WSDL, UDDI, BPEL4WS, and WS-CDL) [1].

Today, the criteria of flexibility, adaptability and security become the essential criteria in the development of service- oriented applications [2].

The BPEL language, often used to model the business processes, is a static language, not adaptable, and it does not provide any support for the specification of either authorization policies or authorization constraints on the execution of activities composing a business processes [3].

The ECA Rules (Event-Condition-Action) are appeared firstly in the active databases. They have the following general semantic: when an event occurs, if the condition is e action is executed. They are subsequently used in other application areas, namely: the active warehouses, the active networks and the business process [1].

The use of ECA rules is very appreciated to insert the flexibility into the process modeling [4]. Generally, they are widely adopted to model the business rules. Furthermore, separation of concerns provides a way to separate development of the functionality and the crosscutting concerns (e.g., context, security). Its advantages are: transparency, evolution, understandability and scalability [2]. This principle became one of the basic principles in software engineering [5].

To incorporate flexibility and adaptability into a business process design, and benefit of the advantages of separation of two concerns: security and interaction in business process modeling, we propose, in this paper, a new rule based model that wants to improving the flexibility, adaptability and verification of business process.

We inspire our approach of the relative works [6][7]. The first work uses a formalism named ECAPE to model flexible business processes, but it does not adopt the separation of concerns. The second work adopts the separation between the concerns security and interaction but it does not take into account the parameter Time in its formalism of modelling of business process. In our work, the parameter time appears important, in particular for the security concern, to avoid the inactivity of the processes and the intrusions.

In our approach, each business process is specified by a set of rules which use our formalism CECAETE (Concern, Event, Condition, Action, check Execution, Time, else Event) based on business rules. We describe our approach follows an example of car rental. In occur, the graph of rules will be built and analysed to verify and handle exceptions in the rules based business process.

This work is organized as follow:

In the second section, we present our new rules based model. It describes the steps of our approach according to an example. Third and the four sections are devoted to the discussion of handling the exceptions in the process by analysing the graph of rules and by the formal verification. The Section 6 presents the related works. Finally, some concluding remarks and further required extensions of this work are given.

2. Multi concern Rules Based Modelling of flexible Business Process

Introduction

ECA formalism has been adopted by many languages for rules based modeling of business processes. This is justified by the fact that this formalism allows to integrate all types of business rules (constraints, derivation, production, and transformation). Business rules are seen as policies, laws and know-how to deal in any business.

The capture of business rules as ECA rules with the separation of concerns have many advantages [2],among others, (1) the inherent ability of adapting any concern rules before imposing them on running services or components; (2) the promotion of understandability of each concern in isolation and then the study of the coherent composition.

To get the flexibility, adaptability and the separation of concerns: interaction and security, we propose the formalism CECAETE. This formalism is described as follows:

TYPE Concern On Event IF Condition DO Action

POINT Execution check

T Time

Post Event Event.

Its semantics is: for each concern (C), when the event (E) occurs, the activated rule evaluates the condition(C). The condition is either a Boolean expression or a SQL query on the database. If the condition is satisfied, the action (A) is executed. The time (T) indicates the time of execution, or the earlier time of execution or the later time of execution. The event takes place after the activation of the rule, to activate the other rules or to send a message.

b. Example**:**

To understand our approach, we chose an example of car rental, as follows:

Having received customer's request, the system of rent of cars treats the request by the calculation of the Initial amount of rent, more to find a closest rental agency. When these two procedures end, an estimated bill is sent to the customer. In case of acceptance, the customer has to sign a rental agreement, then an bill final is emitted to the customer. Finally the bill is registered.

Constraints that exist in this scenario: 04 security constraints and 01 interaction constraint. The security constraints are:

1) The client must be authenticated in the system of car rentals.

2) The client must be authenticated in the system of the bank.

3) If the rental period is exceeded a certain day "d", the customer must be authorized by the regional chief of the agency.

4) Before receiving the final bill, the customer signs numerically the lease.

The constraint interaction is that the amount of the customer must equal or exceed the amount of the bill amount + an amount "m" (deposit amount). The **fig 1** show The modeling of this example.

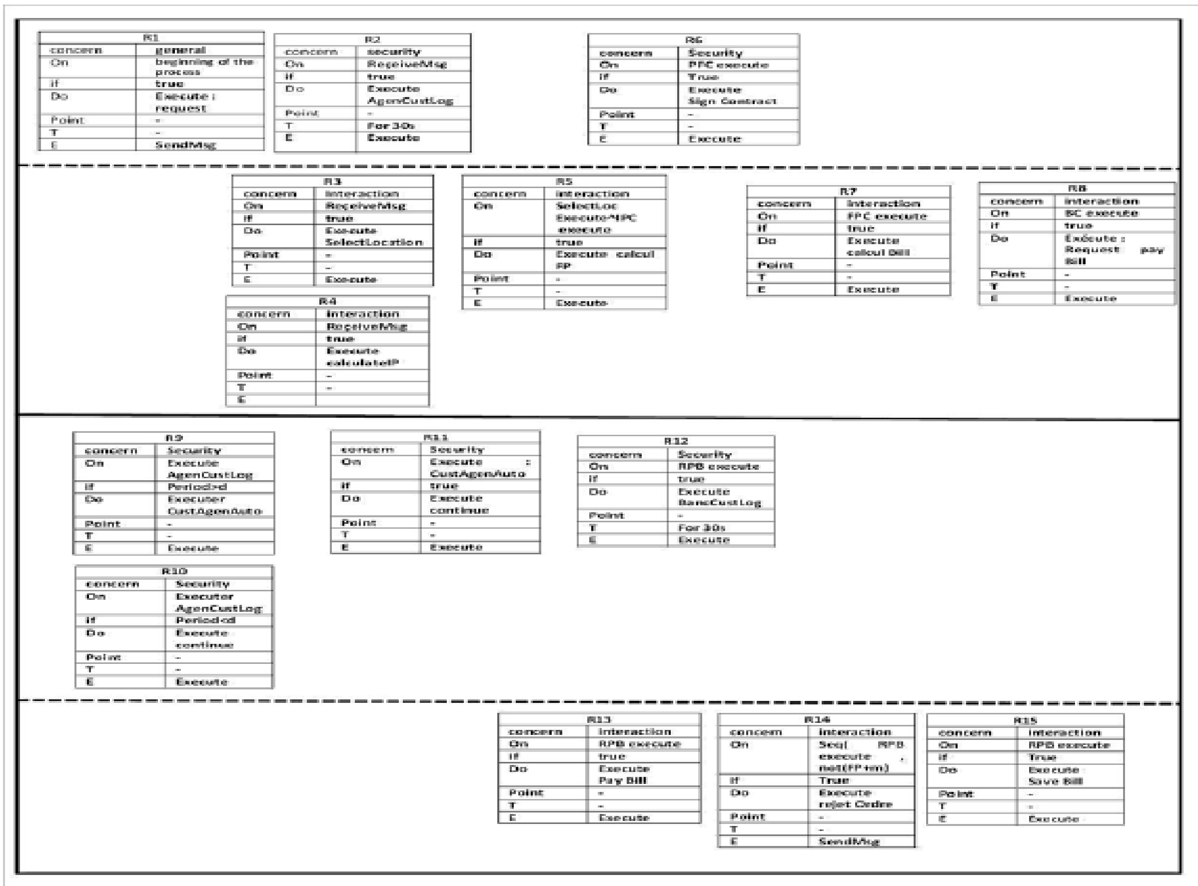

Fig.1 les règles CECAETE de location de voiture

The model above represents the business process management of car rental as a set of rules CECAETE. Business rules are governed separately on two concerns or views. The security view and the interaction view. The separation of concerns promotes the understandability of each concern in isolation.

For example, the rules R2, R6, R9, R10, R11 and R12 are of security concern that governs security constraints. These rules can be modeled by an expert of security, independently of the other concerns. Other rules can be modeled by an expert in the functional concern.

In this model, the rules R2, R3, R4 have the same event to activate witch is the beginning of the process.

The process is started when the arrival of an event (eg clicking on the button "Rent"). However, they cannot be activated at the same time, because they are of two different concerns. The security concern has a higher priority. Accordingly, the rule R2 is activated before the rules R3 and R4. Moreover, the rules R3 and R4 cannot be activated if the rule R2 is not activated successfully or not validated.in case of success, the rules R3, R4 are executed at the same time because they are of the same concern.

3. Exception handling of business processes:

Exception handling of business processes consists in discovering the functional errors on the process and the risks in changing of rules. These risks may be exceptions raised at run time like infinite loop and process non-repudiation, services deny.

To ensure reliability of business processes for the treatment of exceptions, we try to identify exceptions at modelling and at runtime. The detection of errors is too early useful for designers to verify their modelling at high-level. However, the identification of the functional errors must have a state of process data and a scenario execution. But it is often difficult to get this information when modelling [5]. According to the work [6] [7], we draw a graph rule-based causal relationships between these rules. In simple terms, a rule A causes a rule B if A produces a triggering event B or B does not run if A does not run correctly. For our example, the graph becomes:

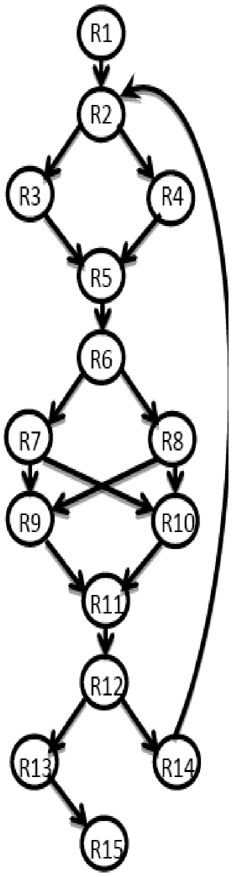

Fig 1. . Relationships graph execution

During the modelling, if we detect a cycle in the graph, it means there is the risk of live lock. We annotate this exception on the attribute CheckPoint. As for the security concern, the expert can annotate the rules that have more risk of intrusion or non-repudiation, as the rules of authentication, authorization and of the digital signature. During the execution, the system executes a specific treatment for the annotated rules, to verify the presence of the exemptions and launch the necessary treatment. In our example, the rules are annotated:

R2	
concern	**Security**
On	**ReçeiveMsg**
if	**True**
Do	**Execute AgenCustLog**
Point	**Risk of intrusion**
T	**For 30s**
E	**Execute**

R12	
concern	**Security**
On	**RPB execute**
if	**true**
Do	**Execute BancCustLog**
Point	**Risk of intrusion**
T	**For 30s**
E	**Execute**

R14	
concern	**interaction**
On	**Seq(RPB execute , not(FP+m)**
if	**True**
Do	**Execute rejet Order**
Point	**LiveLock**
T	**-**
E	**SendMsg**

R6	
Concern	**Security**
On	**PFC execute**
If	**True**
Do	**Execute Sign Contract**
Point	**Non-repudiation**
T	**-**
E	**Execute**

Fig 3. Rules Marked

4. Formal verification of CECAETE Rules based process

Petri net is widely applied in the verification of business process modeling. To do a formal verification of the CECAETE Rules based business process, we found that the ECATNets[8] is the most appropriate Petri net to model it. EcatNets has conditions before and after to fire transitions and it is flexible.

Extended Concurrent Algebraic Term Nets (ECATNets for short) are a kind of high level algebraic nets combining the expressive power of Petri nets and of abstract data types [8,9].

ECATNets semantics is defined in terms of rewriting logic (RL). [10]. Such semantics provides a sound basis for rigorous verification of system properties. RL has been proved very appropriate for dealing with concurrent systems.

Further strengths is its practicability through the efficient MAUDE language [11].

The follow figure shows the EcatNet

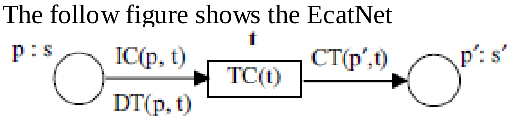

Fig 4 A generic representation of an ECATNet

P is a finite set of places.

T is a finite set of transitions

IC is Input condition, for a given transition t, the expression IC(p, t) specifies conditions on the marking of an input place p for the enabling of t.

DT is Destroyed Tokens. The expression DT(p, t) specifies the multiset of tokens to be removed from the marking of the input place p when t is fired.

CT is Created Tokens. The expression CT(p', t) specifies the multiset of tokens to be created in the output place p' , when t is fired.

TC is Transition Condition. The expression TC(t) is a Boolean term which specifies an additional enabling condition for the transition t. TC(t) specifies some conditions on the values taken by local variables of t (variables related to the all input places of t). Note that when TC(t) is omitted, the default value is the term True.

An interesting feature of ECATNets is that there is a clear distinction between the firing condition of a given transition t and the tokens which may be destroyed during the firing action of t (respectively specified via the expression IC(p, t) and DT(p, t)). A transition t is fireable when several conditions are satisfied simultaneously: (1) Every IC(p, t) is satisfied for each input place p of t. (2) The transition condition TC(t) is true. When t fires, DT(p, t) tokens are removed from the input place p and simultaneously CT(p' , t) tokens are added to the output place p' .

In Maude object states in are conceived as terms — precisely as tuples of the form Id : C|at1 : v1, .., atk :vk. In this tuple : Id stands for object identity; C identifies an object class; and at1, ..., atk denote attribute identifiers with v1, ..., vk as current values. Messages are regarded as operations sent or received by objects, and their generic sort is denoted Msg. Object and message instances flow together in the so-called configuration, which is a multiset, w.r.t. an associative commutative operator denoted by '-- ', of messages and (a set of) objects. The effect of messages on objects is captured by appropriate rewrite rules [2].

The verification process consists of the following steps:

1. Acquisition of business process based on rules.
2. Transformation the CECAETE rules to ECATNets.
3. Description of ECATNets in rewriting logic.
4. Verification of the generated description with the MAUDE tool.

After a careful analysis of the properties of CECAPNETE and ECATNETs and inspired from [12], we propose a transformation of the CECAETE to ECATNETs shown in the following table:

CECAETE		ECATNET
Type	Concern	Type of marking
On	Evnt	Event
If	Condition	Input condition (IC)
Do	Action	Destroyed token (DT)
Time	Time-condition	Transition condition
Point	Execution-check	-
Post-event	Trigger	Created token (CT) +

Fig 5 Transformation the rules of CECAPNETE To ECATNETs

The formal verification of CECAETE rules based business process consist to verify the functional exceptions and properties like: Deadlock, Live lock, Boundness and controllability.

The detail of the verification is not given in this paper.

5. Related work:

In [13], the authors of this paper made a comparison between two approaches of modelling of business processes. Graph-Based Process Modelling Approaches and Rule- Based Process Modelling Approaches. In the 1st approach, the activities are represented as nodes and the dependencies between them such as arcs. In the 2nd approach, process logic is abstracted into a set of rules, each of which is associated with one or more business activity, specifying properties of the activity such as the pre and post conditions of execution.

After a comparative study between both approaches [13], the authors deduce that Rule-Based Approaches are more flexible and more adaptable.

In [7], the authors of this article proposed a multi concern approach to model flexible business process. But, they didn't capture the time attribute and they haven't done a formal verification of the modelling.

In [6], the authors defined a new a framework of the modelling business process it's the model "ECAPE" with the aim of transforming a process in a graph of rule which can be analysed in term of reliability and flexibility. But in this model, they used a single view and did not adopt the separation between concerns.

6. Conclusions and future work

In this paper, we proposed a new model for describing business processes rules-based ECA, which improves flexibility and adaptability of business processes, enjoying the benefits of the separation of two concerns: Security and Interaction. The approach is fully illustrated using an example of a rental car. In future, we will experiment this approach with more complex examples of E-commerce.

Using the aspect programming, we can extend our work by
a transformation of the modelling to an aspect oriented code.

7. REFERENCES:

1. OASIS SOA Reference Model TC : http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=soa-rm.

2. Nasreddine Aoumeur ,Kamel Barkaoui ,Gunter Saake A multi-dimensional architectural approach to behavior- intensive adaptive pervasive applications, Proceeding ISWPC'09 Proceedings of the 4th international conference on Wireless pervasive computing, IEEE Press Piscataway, NJ, USA ©**2009**

3. Mohamed Boukhebouze, Youssef Amghar, Aïcha-Nabila Benharkat, and Zakaria Maamar. Towards Self-healing Execution of Business Processes Based on Rules. Springer- Verlag, **2009**.

4. G.-J. Ahn, R. Sandhu, M. Kang, and J. Park. Injecting RBAC to secure a web-based workflow system. In Pro- ceedings of the 5th ACM Workshop on Role-Based Access Control, pages 1–10, **2000**

5. Raman Kazhamiakin, Salima Benbernou, Luciano Baresi, Pierluigi Plebani, Maike Uhlig and Olivier Barai. Adaptation of Service-Based Systems Service Research Challenges and Solutions for the Future Internet, Lecture Notes in Computer Science, **2010**,

6. M Boukhebouze, Y. Amghar, A. Benharkat, Z. Maamar. A Rule-based Approach to Model and Verify Flexible Business Processes. : International Journal of Business Process Integration and Management: IJBPIM (). **2010**

7. Khadhir Bekki, Hafida Belbachir. A flexible way for adaptive secured service-oriented business processes modelling. Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications, ISWSA '11.

8. M, Bettaz and M, Maouche, "How to Specify Non Determinism and True Concurrency with Algebraic Terms Nets". In. LNCS 655, Bidoit, M., Choppy, C. (eds.), Springer-Verlag, Heidelberg, 1992, pp. 164 -180.

9. K. Barkaoui and A. Hicheur. "Towards Analysis of Flexible and Collaborative Workflow Using Recursive ECATNets". (CBP'07), LNCS 4928, Springer Verlag, 2007, pp. 206-217.

10. J. Meseguer. Conditional rewriting logic as a unified model for concurrency. Theoretical Computer Science, 96:73–155, 1992.

11. M. Clavel, F. Duran, S. Eker, P. Lincoln, N. Marti-Oliet, J. Meseguer, and C.L: Talcott.All About Maude - A High- Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic. Lecture Notes in Computer Science (springer), 4350, 2007.

12. Khadhir Bekki, Hafida Belbachir: A Flexible Integration of Security Concern in Rule Based Business Process Modeling., Proceedings of the 4th International conference on Web and Information Technologies, ICWIT 2012, Sidi Bel Abbes, Algeria, April 29-30, 2012

13. Ruopeng Lu and Shazia Sadiq, « A Survey of Comparative Business Process Modeling Approaches », In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 82–94. Springer, Heidelberg (2007).